ALGORÍTMICA NUMÉRICA 12 Enero 2012

Examen Computacional 3: Enunciado

Ejercicio 1

1. Dibujar una gráfica para comprobar que la función $f(x)=x^3+4x^2-10$ tiene una única raíz real s. Buscar un intervalo [a, b] en el que se encuentre la raíz s.
   Aplicando el método de Newton para calcular las raíces de f(x) obtenemos la siguiente iteración:
   $$x_{k+1}=x_k-\frac{f(x_n)}{f'(x_k)}=x_k-\frac{(x_k)^3+4(x_k)^2-10}{3(x_k)^2+8x_k}=\frac{2(x_k)^3+4(x_k)^2+10}{3(x_k)^2+8x_k}$$
   Escribir un script que implemente 10 iteraciones de la fórmula:
   $$x_{k+1}=\frac{2(x_k)^3+4(x_k)^2+10}{3(x_k)^2+8x_k}$$
   con x_0=a. En cada iteración volcar por pantalla una información similar a la siguiente:

   Iteración: 1 valor x_k: 1.4545454545454546 valor f(x_k): 1.540e+000
   Iteración: 2 valor x_k:

   ¿Cuál es el valor de la raíz s?.
   ¿Cuántas iteraciones, n_iter, han sido necesarias para alcanzar la precisión de la máquina?.
   ¿Cuántas cifras significativas se han obtenido en la iteración 1?, ¿y en la iteración 2?, …. ¿y en la iteración n_iter?.
   ¿Cuál es la velocidad de convergencia (lineal, cuadrática,…)?. Justificar.

2. Dibujar una gráfica para comprobar que la función $g(x)=e^x-2x^2$ tiene 3 raíces reales (s_1 negativa, s_2 y s_3 positivas).
   Utilizar un procedimiento similar al del ejercicio anterior para obtener la iteración resultante de aplicar el método de Newton para calcular las raíces de g(x).
   Escribir un script que implemente la iteración obtenida, arrancando en x_0=0, volcando por pantalla la misma información del apartado anterior.
   Repetir la ejecución arrancando en x_0=1 y en x_0=3.
   ¿Cuáles son las 3 raíces de g(x): s_1, s_2, s_3?.
   ¿Cuántas iteraciones son necesarias para alcanzar la precisión de la máquina?.
   ¿Cuál es la velocidad de convergencia del método?.
   A partir de la gráfica de g(x), justificar porqué cuando arrancamos en x_0=0 el método aproxima la raíz s_1, mientras que cuando arrancamos en x_0=1 el método aproxima la raíz s_2. ¿Dar una aproximación del intervalo de convergencia de la raíz s_2?

Ejercicio 2

1. Ejecutar el código N=6;J=ones(N);H=3*eye(N)-triu(J,-1)+triu(J,2) para definir la siguiente matriz H de dimensión 6:

```
   A =

  2    -1     0     0     0     0
 -1     2    -1     0     0     0
  0    -1     2    -1     0     0
  0     0    -1     2    -1     0
  0     0     0    -1     2    -1
  0     0     0     0    -1     2
```

   Ejecutar [L,U]=lu(H). Comprobar que las matrices L y U son una factorización LU de la matriz H.
2. Codificar la función X=inversa(H) que a partir de la matriz H calcula su matriz inversa X resolviendo los 6 sistemas lineales $Hx_1=(1,0,0,0,0,0)^T,\ldots,$ $Hx_6=(0,0,0,0,0,1)^T$, utilizando la factorización LU.
   Podeis utilizar la siguiente plantilla:

```
function X=inversa(H)
%Calcula X la matriz inversa de H
[L,U]=lu(H);  % Factorizamos H=L*U
 b=eye(6);x=zeros(6);y=zeros(6,1);
 for k=1:6
                % Resolver el sistema L*y=b(:,k) con la función solve_U
                % Resolver el sistema U*x(:,k)=y con la función solve_L
 end
return
```

   Comprobar la función inversa. Calcular la precisión numérica obtenida que viene dada por $\|X*H-I_6\|$ siendo I_6 la matriz identidad de dimensión 6. El comendo norm calcula la norma matricial.
3. Comparar las prestaciones numéricas de la función inv de Matlab con nuestra función inversa.
   Definir una matriz H de dimensión 500x500. Calcular su inversa con las dos funciones (inv e inversa), medir la precisión numérica y añadir tic/toc para medir el tiempo de ejecución.
   ¿Cuál de las dos funciones es mas precisa numéricamente? ¿Cuál es mas rápida?. Comparar numéricamente la precisión y el tiempo de ejecución.